Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A FULLY DISTRIBUTED SECURE APPROACH USING NONDETERMINISTIC ENCRYPTION FOR DATABASE SECURITY IN CLOUD

SRINU. BANOTHU[1*], A. GOVARDHAN[2], AND KARNAM. MADHAVI[3]

[1]Research Scholar, JNTUH, Asst. Prof., Dept of CSE, Vignan Institute of Technology and Science, Hyderabad, India
e-mail[2]Professor,Department of CSE, JNTUH University,Kukatpally, Hyd. Telangana, India
[3]Professorepartment of CSE, GRIET,Bachupally, Hyd. Telangana, India

[*]Corresponding author: Srinu Banothu

E-mail: srinub1307@gmail.com*, govardhan_cse@jntuh.ac.in, bmadhaviranajan@yahoo.com

ABSTRACT

Database-as-a-Service is one of the prime services provided by Cloud Computing. It provides data storage and management services to individuals, enterprises and organizations on pay and uses basis. In which any enterprise or organization can outsource its databases to the Cloud Service Provider (CSP) and query the data whenever and wherever required through any devices connected to the internet. The advantage of this service is that enterprises or organizations can reduce the cost of establishing and maintaining infrastructure locally. However, there exist some database security, privacy challenges and query performance issues to access data, to overcome these issues, in our recent research, developed a database security model using a deterministic encryption scheme, which improved query execution performance and database security level. As this model is implemented using a deterministic encryption scheme, it may suffer from chosen plain text attack, to overcome this issue. In this paper, we proposed a new model for cloud database security using nondeterministic encryption, order preserving encryption, homomorphic encryption and database distribution schemes, and our proposed model supports execution of queries with equality check, range condition and aggregate operations on encrypted cloud database without decryption. This model is more secure with optimal query execution performance.

Keywords: *Cloud Computing, Database-as-a-Service (DBaaS), Cloud Service Provider (CSP), Database Security, encryption.*

1. INTRODUCTION

Cloud computing is a technology, provides various remote services on a pay and use basis. The cloud services are broadly categorized into three types: 1) Software-as-a-Service (SaaS) 2) Platform-as-a-Service (PaaS) 3) Infrastructure-as-a-Service (IaaS), the prime example of SaaS is Database-as-a-service (DBaaS), it allows organizations and end users to easily outsource their databases and computations and access data whenever and wherever required through any device connected to the internet. DBaaS provides organizations with unlimited data storage services cost-effectively with higher availability and easy deployment.

1.1. DBaaS Architecture

The cloud database setup is shown in Architecture that the cloud database is hosted by various cloud service providers and available over public cloud network to be rented out, use it as a service. The architecture of DBaaS is shown in **Fig. 1**, Cloud databases offerings bundle together a package of database management services, where organizations no need to deploy and manage their database servers and infrastructures, databases are hosted and managed by a third party and accessed by users on the cloud across the globe on pay and use basis. In which any organization or individual user run the application and upload or retrieve the data from cloud databases.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

1.2. Benefits of DBaaS

There are many factors, demanded need for cloud database services and the following are benefits of it.

- Highly Scalable: Infinity data storage capacity
- Cost-Effectiveness: this is a major advantage, only pay for what we use, cost of hardware and networking also eliminated
- For businesses struggling to manage their data, the cloud can provide a low cost alternative to investing in the infrastructure to manage it all at their sites
- For DBaaS, the organization pays for what it uses and time it uses, this is also a big advantage to the cloud database service users

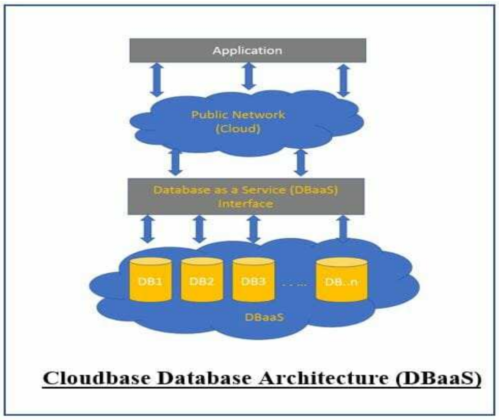

Fig.1. Database As-A-Service Architecture

Along with benefits, there are some challenges also in DBaaS, the biggest challenge is data security and privacy in the cloud environment. Now a day's most organizations or individuals are outsourcing their databases to the cloud environment, the amount of sensitive data stored in the cloud is increasing day by day; hence it should be protected from malicious parties. It introduces new challenges regarding database security and privacy. The major threats to user data are 1) protecting data from external attackers 2) protecting data from cloud service providers. Security and privacy to cloud databases can be provided using 1.Data Distribution Approaches and 2.Data encryption techniques. Authors contributed their work to protect cloud databases from malicious attacks, few of them used data encryption methods and others used data distribution methods. In this paper we proposed a new model for cloud database security using a combination of data encryption and data distribution approaches, the basic idea of our proposed method is initially all the tuples of a relation are encrypted using AES-CBC-256 algorithm using random initialization vector and a secret key, it outputs nondeterministic cipher blocks for the same plaintext block and then the relation is partitioned vertically with selected columns into two or more fragments and store these fragments of tables into different database instances of the same cloud environment and also one additional index column is added to each table fragment, index column is encoded with the hash function, to retrieve the tuple values, a query will be sent to all database instances, processed on encrypted database tables, the result returned to the user is in an encrypted format, the user will decrypt the result using a secret key. Our proposed method addresses the data confidentiality and availability issues of the cloud database and reduces the query processing time to access the data from the cloud database.

The remaining part of this paper discusses the concepts: section II covers related work, section III explains the proposed model methodology, section IV Results and Implementation and section V covers conclusion and future scope.

2. RELATED WORK

In 1978,Ronald L.Rivest et al.[1] Introduced encryption is a well-known technique for preserving the privacy of sensitive data, and also presented the limitations of the model. The authors also demonstrated an application that how to protect and access a small loan company data, which uses a commercial time-sharing system to store the records of loan company data bank. For data encryption, privacy homomorphism techniques are used in their model. Also introduced some sample privacy homomorphism, some of them is weak cryptographically and a "chosen cipher text" attack may break them.

In 1981,George davida et al.[2] proposed a model for database encryption using sub-keys, the basic idea of this scheme is database is encrypted using the Chinese Remainder theorem which satisfies some of the required properties such as security, speed, record level encryption, and attribute based data access by users using distinct sub-keys.

In 2002, HakanHacigumus et al. [3] proposed a model to address the problem of data to be protected from database service providers and also proposed a technique to execute SQL queries over the encrypted database. Introduced a database encryption algorithm for the full SQL query. The basic concept of the algorithm is first, every tuple of a relation must be encrypted with a secure encryption algorithm, then perform weak encryption to the some of the attribute values, it is performed by mapping attribute plain text values into a certain interval and encrypting that interval by a secrete permutation. Then these weakly encrypted attribute values are appended to the actual cipher text. Therefore different plaintext values may be mapped to the same cipher text. But the information available in the plaintext may be destroyed a little bit, but this is not the same as in an ordinary encryption scheme. With small post-processing, the remaining information (like the number of tuples of the table, or which tuples have similar values in which secret attributes) is sufficient to query the encrypted database.

In 2006,Evdokimov et al. [4] introduced a new security definition for Database Privacy Homomorphism, the idea is the construction of database Privacy Homomorphism based on a searchable encryption scheme, in this scheme initially, create some words, those are strings of the same length and then identify the attributes of the relation. Then bijectively convert the tuples of the given relation to the sets of words or documents. The number of words in each document is the same as the number of attributes in the relation. The globally fixed word length is equal to the length of an attribute identifier plus the length of the longest attribute value. Then documents are stored on a remote server by encrypting using a searchable encryption scheme. To apply an exact select query on the encrypted relation, queries will be converted into the search operation and processed as a search operation, returning a set of encrypted strings. The strings are then decrypted and converted into the corresponding tuples. It is a generic construction for a database PH, this can be proved to be secure in a relaxed way, but still requires rigorous and plausible sense under widely accepted cryptographic assumptions.

In 2012, DongxiShenluWang et al.[5], contributed work for secure query processing over the encrypted database, it is named as programmable order-preserving indexing scheme. This scheme is built over simple linear expression of the form a*x + b, the form of expression in public, ‘x’ is the input value and coefficients ‘a’ and ‘b’ are kept secret (not known to attackers). By using linear expression the indexing scheme maps input value ’x’ to a*x + b + noise, where noise is a random value. If noise is carefully selected then the order of input values is preserved. This indexing scheme allows the programmability of basic indexing expressions, in which users can select different linear expressions for different input values for indexing input values. Programmability improves the robustness of the scheme against brute force attacks since there are more indexing expressions. This scheme is used to process range queries over the encrypted database and it only depends on linear expression, so that it is easy to understand by the users. The problem with this scheme is more processing overhead as different linear expressions are used to create indexing for different input values.

Authors in [6] proposed a model for cloud database security in Database-as-a-Service; it provides data privacy and security using data distribution techniques instead of data encryption. This technique is used by the existing netDB2 service. It is based on the multiple service providers and secret sharing algorithm, the basic idea of secrete sharing method is to distribute data to multiple servers to ensure the privacy of user queries. If the user wants to outsource data from a data source (D) to database service providers (DBS1,DBS2,……, DBSn), data is partitioned into *n* shares and *n* shares will be stored in *n* DBS. If the user wants to retrieve the data from DBS, the query will be sent to all DBS and data received from all DBS will be merged and the result will be sent to the user. To reconstruct secrete value Vs at data source D, the knowledge of any K can refer to Vs besides some secrete information X that is known only to the data source. Therefore with the full knowledge of (K-1), DBS will not have any knowledge of Vs, even if X is known to them. In this model, data source (D) selects a random polynomial equation q(x) of degree (K-1), where the constant is Vs. Each DBS has constant Vs and X which is a set of *n* random points. The problem with this model is the availability of all DBS. If any one of the service provider’s server is down, data cannot be retrieved and another problem is the computational complexity of *n* random polynomial equations for different *n* values. Another issue identified in this model is authors only considered numeric data for encryption, doest talk about non-numeric data.

In 2017, authors in [7] recently proposed a model for cloud database security to improve security level, this model provides security to a cloud database using a combination of data distribution and data encryption techniques. This model uses two types of clouds one is master cloud and another is slave cloud, the master cloud stores the entire database encrypted using some encryption algorithms and the slave cloud stores the extended columns (i.e. vertically fragmented columns)of relation. Keys are not revealed to master cloud service providers. In this model, when a relation for master cloud is created, one additional column is added for storing the indexes of each tuple in that relation. The index column stores the indexes of the tuple in plain text format so that the user can query the relation through that index to access the desired tuples. The index is replicated in each fragment stored in the slave cloud. Here master cloud is a private cloud because it is available within the enterprise limits, it acts like a proxy server, the task of the proxy server is to create relations, insert, delete, encrypt, decrypt and process the queries. The problem with this model is that since it maintains a private cloud locally in the enterprise environment, the same infrastructure has to be established and maintained locally as public cloud, so there is no benefit of cloud service reflected. Another issue is all slave cloud servers must always be available to retrieve the data by users. If any one of the slave cloud servers is down data can't be retrieved, so losing on-demand cloud service. And also increases query processing time to access the data as the query has to be split forwarded to all the slave clouds.

In 2020, Authors in [32] proposed a model for database security in cloud known as Proficient Security over distributed Storage: A method for data transmission in cloud. Authors focused on issues of data security on multiple clouds. Proposed model partitioned data into two major types such as normal data and sensitive data again sensitive data is further partitioned into two parts. Every individual part is enciphered and distributed over multiple clouds and normal data is placed over a single cloud in cipher text format. While decryption, sensitive data is retrieved from multiple clouds and combined. The proposed model also tested against various attacks and also proved that model is resistant to pollution attack, known plain text attack and key attacks. But still this method suffers from availability property of information security.

3. PROBLEM STATEMENT

We propose a model for the outsourced database security and availability problem in Fig. 1. The proposed model involves 3 entities: the data owner, the cloud service provider (CSP), and a group of data users. Initially the data owner creates a database, outsources it to the CSP, and shares it with the group of data users. Along with the shared database, some metadata also stored in the data owners local system to enable database security. Every user in the group is allowed to access the shared database and to modify and search the records in the shared database. When a data user submits a query request to the CSP, the CSP returns the corresponding records to the user with and a proof for integrity verification. Then the user verifies the correctness and completeness of the query results using the proof. If the results are correct, the user decrypts the records at user machine and outputs Accept, otherwise the process terminates and Reject outputs.

4. PROPOSED MODEL

In our model, the basic architecture of secure Database-as-a-Service (DaaS) model is shown in **Fig. 2**, in which the Data owner outsources databases encrypted using the secure secret key to CSP, then data owner shares the secret key securely to data users through a secure channel, CSP stores the encrypted database, and process the query requests received from data owner or user on an encrypted database. Data users send the query to process on the encrypted database without decryption to CSP, and then CSP processes the query and returns the results to the data user.

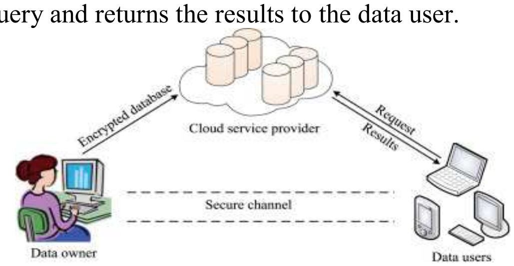

Fig. 2. Basic Architecture Of Secure Database-As-A-Service (Daas) Model.

In our proposed model we are using the following database encryption schemes in combination with the data distribution approach for enhancing database security in the cloud.

4.1. Encryption Schemes

Traditional Encryption schemes provide strong security guarantees, such as symmetric encryption algorithms like AES, DES, etc. However, when these traditional encryption schemes are used for database encryption, leads to unavoidable database search and processing problems, those are mainly three types: Equality check, Order Checking and Computability.

Equality Checking: Whenever plaintext data in the database are encrypted using a traditional encryption scheme, the same plaintext blocks may be mapped to

Journal of Theoretical and Applied Information Technology
15th April 2022. Vol.100. No 7

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

different cipher text blocks if different keys or initialization vectors are used. So that it is very difficult to search text.

Order Checking: When numeric data values are encrypted using a traditional encryption scheme, it losses the order of numeric data, so it is very difficult to search order of data, due to this range queries cannot be applied to the database.
Computational problem: When plaintext data is encrypted using traditional encryption algorithms, we cannot perform operations like addition or multiplication on cipher text data. Due to this queries with aggregate functions cannot be applied to the database.

To overcome the above issues, we are using three categories of algorithms in our proposed model, so that we can execute range and aggregate queries on encrypted cloud databases without decrypting the database tables.

- Nondeterministic Encryption
- Order Preserving Encryption
- Homomorphic Encryption

Nondeterministic Encryption: The non-deterministic encryption scheme maps the same plaintext blocks to different cipher text blocks, whenever plaintext is encrypted using traditional encryption algorithms like AES-CBC-256 using a secret key and a random initialization vector value using Cipher Block Chaining (CBC) or Cipher Feedback (CFB) modes. So that it is protected from Chosen Plaintext Attack (CPA). In our proposed model, we used the AES-CBC-256 algorithm for database table encryption, since it is more secure and efficient.

Order Preserving Encryption: The order preserving encryption scheme is used for encrypting numeric data in database relations because it preserves the order of data in the cipher text. For example if v1,v2 are two integer values and if $v1 < v2$, then it holds the order that $Enc(k,v1) < Enc(k,v2)$, where 'k' is a secret key and Enc(k,v1) is the encrypted values of v1 using secrete key 'k'. So that rage queries can be executed efficiently and securely on the encrypted database without decrypting. It avoids the order checking problem. We used the most popular order preserving encryption scheme used in [8-10], in our model for numeric data encryption.

Homomorphic Encryption:
Cipher outputs from homomorphic encryption are more secure and all aggregate operations like sum, sub, min, max and average operations are performed on them without decrypting. For example, if x1,x2 are two plaintext values then $E(k,x1)*E(k,x2)$ is equal to $E(k,(x1*x2))$ and $D(k, E(x1*x2))$ is equal to $x1*x2$, where E(k, x) is the encryption of plaintext values using secrete key 'k' and D(k,C) is the decryption of cipher text C using secrete key 'k'. So it holds the multiplicative homomorphic property. This encryption scheme is designed for executing aggregate SQL queries on cipher text blocks without decrypting them. In our model, we used the homomorphic encryption scheme in [7]. This homomorphic scheme is very efficient

4.2 Proposed model Methodology

First, database relations are encrypted using appropriate encryption schemes, then relations are vertically fragmented with selected columns of relations (i.e. column selection for table partition is based on data sensitivity level) and stored in cloud databases. Two types of databases are used, one is a master database and another is a slave cloud database, master database is used to store the metadata information in the data owner environment, slave cloud databases are used to store the fragmented tables. The metadata in the master database includes the relations with fields like relation name, column names and database name; this is required for data owners and data users for easy retrieval of data from cloud databases. For data encryption, AES-256-CBC uses a secret key and a random initialization vector is used for nondeterministic cipher text blocks so that the same plaintext blocks are mapped to different cipher text blocks using the same secret key. It provides very good data confidentiality and high security to the data with optimal database encryption time, and we used Order Preserving Encryption (OPE) scheme for executing range queries over encrypted cloud database, homomorphic encryption for aggregate query execution over encrypted cloud database and also used the hashing encryption scheme for equality condition checking i.e. blind index, this model is called as fully secure distributed approach (FSDA) using the blind index. The methodology of my proposed model:

- First, additional columns are added in cloud database relations, one for the blind index for equality check, one column for storing the values encrypted using Order preserving encryption for range condition checking and another column for storing domain values

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

encrypted using homomorphic encryption for aggregate query processing.

- The data owner encodes the primary key column domain values of the table using the hash encoding scheme and stores them in the blind index column of the cloud database table, here the SHA-256 encoding scheme is used.
- Then all numeric column domain values are encrypted using the Order Preserving Encryption scheme [7] and stored in additionally added columns of cloud database for executing range queries over encrypted database table without decryption.
- Data owner also encrypts the domain values of columns using a homomorphic encryption scheme on which aggregate queries are to be processed and store them in additionally created columns in the cloud database.
- Finally, the data owner encrypts all the column values of a relation in a database using AES-256-CBC encryption algorithm, a secret encryption key and a random initialization vector (IV), this key is known only to the data owner and it should be securely shared to the data users.
- Then the encrypted database tables or relations are vertically partitioned into two or more fragments with selected columns by considering data sensitivity criteria and also add index column for each vertically fragmented relations, this index value must be replicated in all fragments for the tuple of un-partitioned relation so that the user can retrieve the tuple data easily and reduces the query execution time.
- Uploads the encrypted and vertically fragmented table in multiple database instances of the same cloud service provider environment.
- Data owner maintains metadata in the owner-private environment to know the locations of fragments stored in cloud databases.
- Data owners must authenticate users to perform operations on a cloud database, for authentication users must register with the data owner with their details.
- The data owner will share the user credentials with cloud service providers (CSP), so that CSP verify the user credentials with credentials already shared by the data owner to CSP, if a user is valid then CSP grant permissions to the user to access data.
- Data users can send the query request to the data owner for the metadata information.
- The data user can retrieve the encryption key from the data owner and perform operations on databases like selection, insertion, deletion and updating.
- The data owner or user retrieves the data from the cloud database in encrypted form only and performs decryption at the client environment using the secret key. So CSP doesn't have any knowledge about the data stored in the cloud.
- When the user sends the SELECT query to retrieve the data from the database, it must include the JOIN clause with a predicate on the index column.

So the advantage of our proposed cloud database security model is that 1) it is strongly protected from Chosen Plaintext Attack(CPA) because, in this model we used a random initialization vector in AES-CBC-256 Algorithm for database relation encryption, it maps the same plaintext into different cipher text blocks. 2) As database relations are partitioned vertically and distributed into multiple database instances if the attacker compromises the data in one fragment, cannot get the complete information. The cloud service providers will be unaware of the data stored in the database because all attribute values of records in database relations are stored in cipher text format. 3) Proposed model provides high availability service compared existing approaches in literatures, as in our proposed model database is stored in single cloud service provider's environment instead of multi-clouds.

5. IMPLEMENTATIONS AND RESULTS

5.1 Cloud Computing Tools

For simulation of our model, we designed an application using PHP and My-SQL server, also used HTML for front end design, for this installed the XAMPP tool and application is deployed on local system. Then we created a public cloud computing account at cloud clusters.io, which provides an open-source cloud computing service. We have created a My-SQL server managed with

PhpMyadmin for experimental purposes and then created two slave databases in the cloud for storing fragments of database relation. The configurations of servers created on the cloud are 3(core) Processors,4GB RAM, 100GB SSD. Then run my application on my local system configured with Intel Core i5 processor,10GB RAM and 360GB hard disk space. The network speed is 150mbps.

5.2 Results and Performance Evaluation

For our experimental work, we have taken employee datasets and stored them in cloud databases. For simulation purposes, we have created two database relations on our local system one for storing metadata and another for storing employee records with fields id, emp_id, emp_name, emp_email, emp_salary, and emp_age., this is called data at data owner side, we also created two slave databases in cloud environment each one for storing fragmented relations and also note that we can create more number of slave databases, which may be equivalent to several fields in data owner base table. We have created a fragmented table schema in slave database1 with fields id, emp_id, emp_name, emp_email and blindindex, here the blindindex column is used to store the hash encoded values of emp_id field, and here we have taken emp_id filed as the primary key. Also created another fragmented table in slave database2 with two columns i.e. emp_salary, and emp_age. Then1000 records of employees are encrypted using a suitable encryption scheme and inserted into the cloud databases, the results are shown in figures. **Fig. 3.** shows the data stored in the data owner local system, it is in plain text format. **Fig. 4**. shows the data stored in a vertically fragmented table in cloud database server 1, in which field values emp_id, emp_name and emp_email are encrypted by data owner using the AES-256-CBC algorithm and secure encryption key and stored in cloud, bindex column values are encoded values of emp_id column using an SHA-256 encoding scheme, bindex column values are required for executing select queries with equality check condition on encrypted database relation without decryption in the cloud environment. **Fig. 5.** shows the data stored in a vertically fragmented table in cloud database server2, these column values are encrypted by the data owner using the AES-256-CBC algorithm and secure encryption key and stored in the cloud.

Emp ID	Employee Name	Email Id	Salary	Age
cse_01	Manoj	manoj@gmail.com	24000	34
cse_02	Jasmitha	jasmitha@gmail.com	15000	21
cse_03	srinu	srinu@gmail.com	34000	40
cse_04	padma	padma@gmail.com	18000	32
cse_05	praveen	praveen @gmail.com	29000	39
cse_06	venkat	venakt@gmail.com	40000	40
cse_07	raju	raju@gmail.com	45000	45
cse_08	rani	rani@gmail.com	12000	40
cse_09	sonu	sonu@gmail.com	33000	34
cse_10	JUNAID ASHRAF SOFI	JUNAIDASHRAF 02018@GMAIL. COM	54000	28
cse_100	NANJUTI VARSHITHA	varshitha@gmail. com	108500	53
cse_101	NIDRA UDAYA SREE	nidrapaemaiah12 3@gmail.com	110000	51
cse_102	NIMMALA GANESH	ganesh@gmail.com	111500	49
cse_103	NYAVANANDI SURAJ	suraj@gmail.com	113000	47
cse_104	YELLAMCHERV U ALEKHYA	alekhya@gmail.c om	114500	45
cse_105	PAMIREDDY MONIKA	venkatareddy185 4@gmail.com	116000	43
cse_106	SEEMA TABASSUM	mdjunaid.464@g mail.com	29178	41
cse_107	SHRIYA GURRAM	SHRIYA@GMAI L.COM	26476	39
cse_108	THATIKONDA AKHIL KUMAR	takiridhar1@gm ail.com	67527	60
cse_109	ALAMPALLY NIKHIL	rkumar@gmail.c om	90296	32
cse_11	ANDELA RAHUL	RY1318297@G MAIL.COM	52000	34
cse_110	ALE VINAY	vinay99@gmail.c om	26388	31
cse_111	ALUGULA HARSHA VARDHAN REDDY	mattareddy.alugu @gmail.com	24053	59
cse_112	ARRAGONI BALARAM	balaram99@gma il.com	35635	62
cse_113	AYYALASOMAY AJULA ADITYA SUDEEP SAI	sudeep280700@ gmail.com	75658	45
cse_114	BUJJI ASHRITH	bujji_bnr43@redi ffmail.com	48521	49
cse_115	BOLISETTY NIKHIL GUPTA	NIKHILGUPTA@ GMAIL.COM	20726	28
cse_116	GUNUPUDI VENKATAPAVA NMANIDEEP	manideep pvana@gmail.co m	50676	63
cse_117	BATTULA ASHISH GOPI	gopi99@gmail.co m	60966	66
cse_118	BEERELLI SADASHIVA REDDY	madhukarreddy @gmail.com	48156	37

Fig. 3. Table data in plain text format stored in local system to be outsourced by the data owner

id	emp_id	emp_name	emp_email	bindex
1	aHVPNG5lQXNiMnZkRm1zb2lua2g0UT09Ojol7bXZe9fO2O6N37...	Z2U1NTVydGplRkNGTDIMM3p5MWN3QT09Ojom39JiV35P453fX+...	[BLOB - 56 B]	2249c7793038e087521150f7d271daae7c4758503a1cc18af4...
2	TEovaXA3UjZlYTRQREZ5YkwvOGNuZz09OjrCC/SbsYhn/hyoFj...	T0JvVk5lbEJablp5K1VBcjNDbVNUUT09OjoHQwY5hiK0HGXPRo...	[BLOB - 84 B]	da6a29824d8fa4e085f7f2f20a9e5189abcc041562542a74c4...
3	aTRpVjUvQS85RDhrY1ByaXlHZW9lZz09OjrRjAFmK602m8PkEx...	SkhMNWlDTGVCdXlmaUZMNUtoTEowdz09OjpqJ0vOyzlMmSq5Gw...	[BLOB - 56 B]	fa72f1d957e698798a5ac7d393891506f114fe69635cc7518f...
4	OVpPRnNrTnlxWlhiUHJ6SkZ1UVFvUT09Ojo4Hhkct5fe2E0kBv...	cGsvVUFGSlBuZlNuSElaMzRzMVN1dz09OjqTlgATIJlaRMSbNW...	[BLOB - 56 B]	ef9154ce66887f7bf5dc2e6d907ff408220533 1b306732c057...
5	RGo4TG9yZDlzQWdlT2FiUHFHSkQrQT09Ojo+Xj+SdlAyOuC3Ry...	QWVFa2cvWlUyejhUWEdjNzI3Uk1FQT09OjpQ9M5uRk4qcnVtNU...	[BLOB - 84 B]	d8403ed7d8283adb77554785979bcf78ef73f3cdf09d038bda...
6	bEFHK05GTnl0c29MWmRjMEhOVW40Zz09OjoCx9MWoSoVpJy5vi...	Y2VHMUNQd09uZ282dU9vSFBKWnFudz09OjoYuDE6Kylx7eks1W...	[BLOB - 84 B]	c189f684163bc5969ded01a6cdd9a7d0b50f8f706905b1a573...
7	QmJZL1hQd3h1eDVCdXRReUzY3JlUT09OjpZNltyZ6NrVlBY2x...	Ynpyc1FMdUlyYWJLUmxhVkxyejFJUT09Ojo0CEJ9rcxTbpLM9j...	[BLOB - 56 B]	bae77a56067256dd426e9db3f17857e952aef1eb607e955c08...
8	SW82SDA4V0FnNUJEaEhaYXlUM0dYQT09OjritNKygDf/1OFLXS...	SUFzSzJNSndoN3VJVTNobWtYRkVpQT09OjpWBT7nyF+/zwvNjq...	[BLOB - 56 B]	e565d3350b8ee1fc675ed138f5aee6f1bca4f835b1de3eec2b...
9	eUpGVVREaDJGMllseGNjN3JMTS9QQT09Ojr4xLLAkUJluyh574...	Q3ZqRTczM2tRSHRBeUdzaVRZMkdiUT09OjqYvf1/gryi9r49Nc...	[BLOB - 56 B]	bd48fd43701af942235954ac4f2a442e7113b1c6fd18cfd43a...
10	NXZvRkE5L1BqUW5FSmZpdXAxSUptZz09OjrT6Osc9ax93X9vpH...	QlBCVHcvZVhPQTNEbkdrbFNFWk5rWm41c3c3ZkY0U3MwQ09kTF...	[BLOB - 84 B]	1a13c027d27a6caa73ab4e10a17adc3846f3c6ae3cb4cb33e8...
11	NHp2K2hYOEczdkZaSjJtdnJ3clozdz09Ojq8+SkWB4FrxMvrCn...	MHlyeUt5UnVOcEVlbGVNMHNIZ0JJdHhzZndoZ2J2eXExN096QV...	[BLOB - 84 B]	76ee5a71e3e7df850216cb89ff479833b80bd73e79fec61010...
12	RWJsYnpJbXNuQy84RmltKzNqNWh5dz09OjrjvRie8WQeDbSbOd...	R0VYZkNaZW5XYUZVNDZzV0NpTzZtcGRlMnJYYTZGeDZOMDRvOU...	[BLOB - 84 B]	22508da98d3e43ea4119ace53bbfe8783077e19cf1a0ab1dd6...
13	Tnl2Um9ZcjhucHlUanp4Y2lBRFBMdz09OjqEbeXP4WfxrlMsRB...	VS81RUc4OTUxTEFKa01jMTZZS3JSQT09OjqzNmWDY7eZdTv1Qn...	[BLOB - 84 B]	cd459885d386f402aa2b601a29dd1d789942219999a4b1d17e...
14	ZFR1SEJNWFVldmplZEg1bXNDL1FJZz09OjpicyNYYJQM0+fBSJ...	dGhzWWo0bCtuVXpvdGU4NHlYL3J1eWpnVURIdTV2TGtlNGNNaD...	[BLOB - 56 B]	b6f2d3d50077f8184a19607f55a4aad6fd6d3da830febf6154...
15	VCtuRmFuRmhweUlqdzI4Z3ZXOUhrdz09OjqTUqOjiS/0zC4DVk...	SjJhSGtGVDU5MTUrZlNNZE42eU00V1pkSHBWUlZvdWU3UkZYeT...	[BLOB - 84 B]	d74ba4ff1be7ece9022a9c081900729759c92f5eab22b7238f...
16	T3lza1RTa0Q4bkd0U3l5dTFXdHZadz09Ojo4RF3uuoAJ2h5YRN...	R0M2dE9MODRTWmZ3QitHSHlTajJGYjVlOUUzWm1jT0tSR3J5Nl...	[BLOB - 84 B]	2941eb075e93f6d112886b2875a5fd068e9dbc77df57b019df...
17	aUFNRmJrN3JHVkZrMjNwNnJpbW1qZz09OjodCh2PA9hSeHJd9e...	RUlrU1VzTnh3MkpRM1h0b1BsM2pqZz09OjoYrj/LvQlYz+5d0H...	[BLOB - 84 B]	a755118fa1a1c555739428ab8dc6fd57fa898eb13657dc9b1b...
18	K2xpeGRLbHpXaDlxcWVKcFlOc25pdz09OjpKt9EKaAtwDg/nzd...	bGF5eWZFSHpJdFBPNkRDRTBraGNqQT09OjpaGf4+/5BgzZVxv8...	[BLOB - 84 B]	094dcf2e44c82c97e1222d365678bdea2e3719cc9d5ed44700...
19	WGdTVno5bTErVThndTNRV2hWbjQzQT09OjqAGiZOhYjHsCMJXu...	aDh1K3BManlEaWJaa0lQRllldEQxL0E4THkwRzZPTVVudjhQNW...	[BLOB - 84 B]	f23b23bd700c23bfe878ef4356df8eea08a7f03e1b753f61a2...
20	TnVlaXU1eUw4bjdPZGlQbnY5a1NXdz09Ojpzuxmar7Ml9Jqkm7...	U3ZicE41YThadlBEcExBSktvWW1nZGVUZ1BNa2xoTk4wbG8zWU...	[BLOB - 84 B]	1719f49a773d6ed186f7335e59fb84548ef4f73e7d7a32d67f...
21	TFBZVjMrVVdsSENiR2tpRndUVlNldz09OjpmapspxZ98YFjee+...	TmZQQ3hVTEZYZlhSWVJCYmhjZFdlUT09OjrQafQyM+R8TD8tH+...	[BLOB - 84 B]	e6b1c671e8ad540fe822e847a8e014a1762e217053368b5290...
22	TjZVTFR4cFJXU1pXS0U5Ris3ejVKdz09OjoCS7GONXn/TFx5J9...	bzAwRU9MZ1lMK29yQzJTMzBoRjNXZz09OjrRlPtmFk1aNrpdvM...	[BLOB - 84 B]	a771a0518c0a463793163f06fdf1576641b5ba1003950f44f9...
23	cWlvV2dCY1hlN3JoUFNyN0dmSzVDUT09Ojp8BOCK8yll0rsQJ5...	YzdrNlpPNFRCbC94MUFYVlMySDZrQkFnZVV3eDkvV1lwdGhkYj...	[BLOB - 84 B]	7370782337539a2c3c944491a28326e1c24334190ab478bebf...

Fig. *4. Table data after encryption and fragmentation stored in cloud database server1*

id	emp_salary	emp_age
1	ZXdLMUs0QmpaMWR0QTYzYkZQYXdRdz09Ojobf15VC0TlbRLUnE...	NG1kRWg0TmJnSXlIRnJQNWozOFE1Zz09OjrZDk/V9IU9Nm0P1Z...
2	UjA1aXZ6VG9lcG1Sd24zcDlqb0hXdz09OjqJWL2YlqrNL68OEV...	a1B5eTVzNWRlYm44NklhN0ViYWhyQT09Ojr9LzjuvURwgBTGDj...
3	RXBCUkdDQXNZTWRxY3JsTno1SHVrUT09Ojq/tdVoqqbD3wqNcB...	U3Z2Q3lxaDlsUlBiWjRpUFJzRkZ3UT09Ojp74uhOAI3uVlHjuN...
4	WWRhL29mN2tla1JGb0pTQUFjRXJwZz09Ojr533NgZ3umQSsbNz...	Z240cXlHcVdXRW5qUGUxZFlndmVlZz09OjrQ6spWzlcYnNcl8d...
5	S0lyYlF0WVZpbTVnUXloaUNwd2dyQT09Ojp6OjFPc/0X+Pgq7N...	ZXJsb09PTmh2dXJ0S3NGVVV1T2Q1UT09Ojr3YmbWyl01fV+T+E...
6	QnZWeFV0NDR0K3phZ0J6K2w5LzUvQT09OjpErOHk5U1dOp2x53...	WnlvWGQ5UmtlTXZpRjlpaUthaGpPZz09OjoQSOE+syfP5kpU//...
7	WHZrbFdkT1FyNUdNSmVlZlcwRHFSQT09OjrSXlo1dRfLFkawDk...	Si9SY0FQUEF5SHR6YWVCdDEzWFVtUT09OjrlaYe3CXwyGJVtqf...
8	WitEL0FLOUt2UGVmcWcvRVNsUUdqZz09OjqJuQqc7kbG9iw9fu...	aCtCbEdGRk9kUDVzN1lKdXV3QXpmdz09Ojral8JFkwRfX0UnJN...
9	a3MzZkRYZFpEazBLSDFOcno3NmNMZz09OjqXi1kam/59/PAERg...	ZzF5aU13NFhqMEgwRElubXQyeUdyUT09OjqcQp1I35bnX1+6Po...
10	SFJmWDdCK2JSODJEZnkwZGdGZnFJZz09OjorkFhxmALpVCQTzG...	MHlqQzFSbEExekkrRS9SUDVvMytOdz09OjqZebvXg4uWSnhuZ8...
11	Z3FtSDNrVlJQNGxpbVdPUG1rN0xXdz09Ojrw6RFm6/EcsJ3YGY...	Y2pJZGh4M1k4TU1ZQzloVS9MLytzdz09OjpNVERYcJ+xkZVao8...
12	TUt0enJ0VkxDZDdGdE11NnBjRVcwUT09OjpfQ+SC86yRNSx/jH...	Sk1JWEtUbUQ5NElzV2d6OW03VVRMdz09OjpDmKi8y6S4oZyEU9...
13	ZFg5b1pyWUhkZVZQZWVicWVLTnMvZz09OjrdSZ8B9WXH/G+fta...	ZzNsQ3drZ2JEN3VQNjl0US9xa1BtUT09OjoLQuDod5/R305BQz...
14	R2FKdGtCTldPZXgyMVhCZXJZZ2NaUT09Ojq2KClQWsHq1IA+Pe...	V0xQREpZM1E1VVhqekpXekw5ZHM1Zz09OjpF6PXrviu7fx+gEW...
15	TnRhUG5OOWlUazFmYzI0MC95c1RHQT09Ojqa4BUm8bzT4LFz0a...	THlGU1JpVmNBdHhmMTZtaW95dloyUT09Ojq2a8CcGXHKoQIR4J...
16	WWZKaHNYYTlwck5MV3crd0NDQ3d4dz09OjrgnBRV/XJVY6zrWl...	UFFrTFJSL0RJWHNmYXA4d2VBWW12Zz09OjqMLzC8aNqC9NErqD...
17	dFVkV0ZSN3VScWNZUUdBYis5T0FPUT09OircuPDN7koJFZ+pnt...	VUtMN3pPUnVwanB5WmROMHdLWTBWUT09Ojo+XKDdv81+hcBRqm...

Fig. *5. Table data after encryption and fragmentation stored in cloud database server2*

Fig. 6. shows the emp_age column values encrypted using a homomorphic encryption scheme by the data owner and stored in a cloud database for executing SQL queries with aggregate functions on an encrypted cloud database without decryption. **Fig. 7.** shows the emp_salary column values encrypted using an order preserving encryption scheme by the data owner and stored in a cloud database for executing SQL queries with range conditions on encrypted cloud database without decryption. In our proposed model we have stored these column values in an isolated cloud server, these records are retrieved using the replicated id column values in each fragmented table.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Id	E(n, Age)
1	136018311391535898537933318185370136982049778993833328 652399214608403358299976812372479312758936664981003133 316017503793295980557346019774952873350402853757447193 463275527794175324476994447008000756713262786553296440 407736376032947742317502522281019593803861083096894074 248794846767840601998812891159037972017753529347906670 115102073743113944729055362820067136347267468191965027 289005142623433538872709799727277147845131329074032538 009136166768453973838333485566238182884942021412763215 859726704897933233204382904927145301106934114483463848 224098848245019087316257232974456392637374739704517509 6753997750940098472100 8
2	791607519735294231475143711303095785632318422527954492 161069115623089421103942274293336087614471240170684227 716335055012620178559458983275507035782089047709406767 165431411658476170795204909897531242740693151679774947 740078650422183792624440090233986505542269853441627307 767932145377211816161772545542518136874548413074475961 771789188256775831033343431039311029262976763364901188 873468791944240436392997878070133295011899643544663267 317400411270619847839192845902162486857341637148269681 176586233383112386303223687988049911855921225944064138 646705338392325495886106680769015593187618073326596960 612059110425415107697 6
3	135965436091360496051047231717947893588650105414791463 466233598302573593174949128839503459389132217776453969 260830180796656759711389786357394988544757250277561674 844161586766958852247567506656291787659084389836326466 284563050113181847245156163309892718914015662242634026 041239517657651338956229744001098806450031900314583149 401602978583972237825462555445027591044119926932404486 615063414777977964220405614894526736051967280947626057 089348009974810031376678639192609828087025273186484782 739908571175217702483516734223935742185368730286325527 174607249400567671317778669096219648521994800808688689 5023290872631318249527 0
4	699940473114879191957361236134890553094829055335835746483150 46 947657822376743907752319854946267385788228253066040590349571 50 682200779567809468145625799972263905772239499191256589432807 02 509277225312008993780458954282433303501461278072907247436275 89 064086452680803980676322351346267472058881979489783623984525 69 784569207719659939821424078864345103646070979534995089315281 42 123671833176337395782626909791111671748618015819127479313618 87 170250938882091897571872376989407426479363228997562569743301 75 229066965226483023854924026219212272257828327054213903827093 99 307696305891183711474408279218830601443120083461339541647 5
5	510249839495401698015556322456168607789224738057893661 447092027531049934078956332415773228494814776578921226 979210659113528280430292462767983830261163649148222460 511828506969574881503070927739865797581574312039924997 428997090594472014149607603359932672701812287053738321 621806989335530574115814547390785383026954191449330808 653921424192294473675748290789004935732610378474087803 352114953641186022504653096075601654669076536749857247 078863815953041743544359555366936277019968436540809051 140588783261316348090858483084760954480002391744291394 737097646635114092259119770632389124055150353674103038 2329339437116213028208

Fig. 6. Emp_age field values encrypted using homomorphic encryption

id	OPE(K,Salary)
1	656206284
2	1309397157
3	1634916838
4	980842854
5	1181804128

Fig. 7. Emp_Salary Encrypted Using Order Preserving Encryption

Table 1.Time Delay In Seconds To Encrypt And Insert The Records In Cloud Databases

Time delay to upload Data in Cloud database(Seconds)		
No. of Records	SCA(sec)	FDSA(sec)
106	58.87	87.39
206	114.22	184.69
306	177.63	265.06
406	243.00	349.15
506	297.23	442.66
606	423.89	473.61

We evaluated the performance in terms of a time delay to encrypt and upload the data in cloud databases and to retrieve records from a cloud database and decrypt at the client-side and show the result to the user. The performance of our method is compared with the existing methods. For performance testing of our proposed model, the first 106 records are encrypted and inserted in cloud databases then 206,306,406,506 and 606.

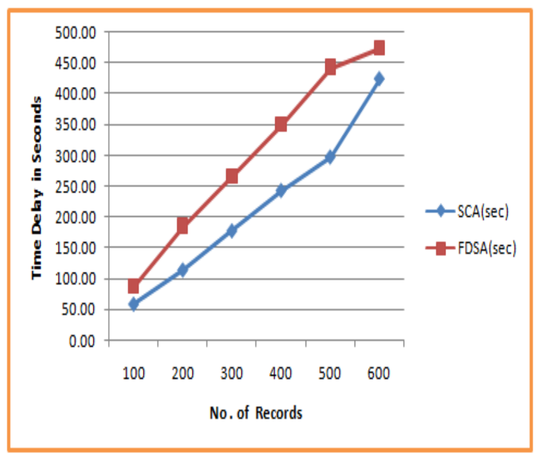

Fig. 8. Time Delay In Seconds To Encrypt And Insert The Records In Cloud Databases Of FDSA And SCA Models

Table 1. shows the performance in terms of the time delay of Secure Centralized Approach (SCA) and Fully Distributed Secure Approach (FDSA) using nondeterministic encryption (i.e. our proposed model)) for encrypting and inserting the records in cloud databases. The data recorded in **table 1** are used to compare the time delay of the existing model with our proposed model for encrypting and executing INSERT/UPDATE query to insert or update data in the cloud database.

Fig. 8. shows that our proposed model performance in terms of time delay for encrypting and inserting the data in cloud database is a little bit slower than the existing model, but as security is a

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

concern our model is more secure than the existing models.

Table 2.Time Delay For Select Query Execution To Retrieve All Records From Cloud Database And Data Decryption At Client Side

Time delay to Retrieve Datafrom cloud database (seconds)		
No. of Records	SCA(sec)	FDSA(sec)
106	1.02	1.04
206	1.89	1.82
306	2.74	2.76
406	3.50	3.03
506	4.29	4.18
606	4.62	5.18

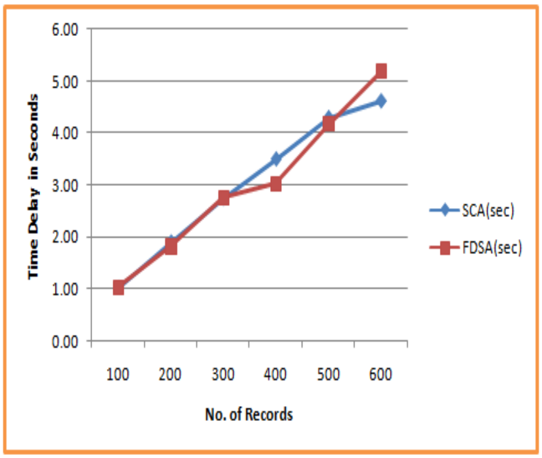

Fig. 9. Shows Time Delay For Select Query Execution To Retrieve All Records From Cloud Database And Data Decryption At Client Side Of SCA And FDSA Models

Table 2. shows the data recorded for testing the performance in terms of time delay for SELECT query execution to retrieve all records from cloud database and data decryption at the client-side with a varied number of records such as 106,206,306,406,506 and 606 records of our proposed model and SCA models.

Fig. 9. shows that our proposed model performance in terms of time delay for SELECT query execution on cloud database servers and decrypting the results returned by the query at the user side is almost the same as the existing model, but as security is a concern our model is more secure than the existing models.

6. CONCLUSION AND FUTURE SCOPE

In this paper, we proposed a model for cloud database security and availability problem using data distribution and nondeterministic encryption approach for Database as a Service in Cloud. The proposed model employed AES-256-CBC algorithm, order-preserving encryption and homomorphic encryption schemes for database encryption and vertical fragmentation technique is used for data distribution. For implementing our model, designed a web application using PHP and My-SQL, we run our application using the XAMPP tool on the local machine and created a database server on open source cloud service provider cloudcluster.io and evaluated the performance of SELECT query execution with equality check, range check predicates in WHERE clause on encrypted cloud databases and measured the time delays to access the records from the cloud. In our research, we have compared the performance of our model with existing state of art methods and found our model is more secure with optimal query execution time and with high availability service. Our future research is to introduce novel methods to further enhance the data security level and data upload performance.

REFERENCES

[1] Ronald L. Rivest Len Adleman, Michael L. Dertouzos "On Data Banks And Privacy Homomorphisms" Massachusetts Institute of Technology Cambridge, Massachusetts Copyright © 1978 by Academic Press, Inc

[2] George I. Davida, David L. Wells, "A Database Encryption System with Sub-keys", ACM Transactions on Database Systems, Vol. 6, No. 2, June 1981, Pages 312-328. https://doi.org/10.1145/319566.319580

[3] Hakan Hacıgum, Bala Iyer, Chen Li, Sharad Mehrotra "Executing SQL over Encrypted Data in the Database-Service-Provider Model", ACM SIGMOD '2002 June 4-6, Madison, Wisconsin, USA Copyright 2002 ACM 1-58113-497-5/02/06 ...}.

[4] Elisa Bertino, and Ravi Sandhu, "Database Security—Concepts, Approaches, and Challenges", IEEE Transactions On Dependable And Secure Computing, VOL. 2, NO. 1, January-March 2005

[5] Evdokimov, S. Fischmann, M. Gunther, "Provable Security for Outsourcing Database Operations", Proceedings of the 22nd International Conference on Data Engineering

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

(ICDE'06) 8-7695-2570-9/06 $20.00 © 2006 IEEE

[6] Sergei Evdokimov, Oliver Gunther, "Encryption Techniques for Secure Database Outsourcing", ESORICS 2007. LNCS, vol. 4734, Springer, Heidelberg (2007) (http://www.springerlink.com/content/978-3-540-74834-2/)

[7] Dongxi Liu, Shenlu Wang, "Programmable Order-Preserving Secure Index for Encrypted Database Query", IEEE Fifth International Conference on Cloud Computing, 978-0-7695-4755-8/12 $26.00 © 2012 IEEE

[8] Dongxi Liu, Shenlu Wang, "DEMO: Query Encrypted Databases Practically", CCS'12 October 16–18, 2012, Raleigh, North Carolina, USA. ACM 978-1-4503-1651-4/12/10.

[9] Lei Xu, Xiaoxin Wu, Hub: HeterogeneoXs Bucketization for Database Outsourcing, Cloud Computing'13, May 8, 2013, Hangzhou, China. ACM 2013 978-1-4503-2067-2/13/05 ...$15.0

[10] Luca Ferretti, Michele Colajanni, and Mirco Marchett, "Distributed, Concurrent, and Independent Access to Encrypted Cloud Databases", IEEE Transactions on Parallel and Distributed Systems, VOL. 25, NO. 2, FEB 2014.

[11] Jiguo Li, Wei Yao, Yichen Zhang, Huiling Qian, and Jinguang Han, Member, IEEE, Flexible and Fine-Grained Attribute-Based Data Storage in Cloud Computing, IEEE Transactions On Services Computing, VOL. 10, NO. 5, SEPTEMBER/OCTOBER 2017

[12] Cheng Guo, Ruhan Zhuang, Yingmo Jie, Yizhi Ren, Ting Wu3, Kim-Kwang and Raymond Choo, "Fine-grained Database Field Search Using Attribute-Based Encryption for E-Healthcare Clouds" J Med Syst(2016) 40:235 , https://doi.org/ 10.1007/s10916-016-0588-0

[13] Md Abdullatif ALzain and Eric Pardede, "Using Multi Shares for Ensuring Privacy in Database-as-a-Service", Proceedings of the 44th Hawaii International Conference on System Sciences – 2011, 1530-1605/11 $26.00 © 2011 IEEE, Pg. No: 1-9

[14] Amjad Alsirhani, Srinivas Sampalli, Peter Bodorik, "Improving Database Security in Cloud Computing by Fragmentation of Data", International Conference on Computer and Applications (ICCA), 978-1-5386-2752-5/17/$31.00 2017 IEEE

[15] Youssef Gahia * and Imane El Alaoui, "A Secure Multi-User Database-as-a-Service Approach for Cloud Computing Privacy", International Workshop on Emerging Networks and Communications (IWENC 2019) November 4-7, 2019, Coimbra, Portugal, Science Direct Available online at www.sciencedirect.com Procedia Computer Science 160 (2019) 811–818

[16] K. Madhavi, G. Ramesh, K. Sowmya, **CICIT,** pp 630-636 (2019).

[17] Srinu Banothu, A.Govardhan, Karnam Madhavi, "Performance Comparison of Cryptographic Algorithms for Data Security in Cloud Computing", Journal of Information and Computational Science, ISSN: 1548-7741, Volume 11 Issue 9 – 2021,Pg. No 1-8.

[18] Srinu Banothu, A.Govardhan, Karnam Madhavi, Performance Evaluation of Cloud Database Security Algorithms, E3S Web of Conferences 309 in *ICMED 2021.*

[19] Bih-Hwang Lee,Ervin Kusuma Dewi, Muhammad Farid Wajdi,Data Security in Cloud Computing Using AES Under HEROKU Cloud,The 27th Wireless and Optical Communications Conference (WOCC2018).

[20] Mr. Manish M Poteya, Dr C A Dhoteb, Mr Deepak H Sharmac,Homomorphic Encryption for Security of Cloud Data,7th International Conference on Communication, Computing and Virtualization 2016.

[21] S.Rajeswari,R.Kalaiselvi, "Survey of Data and Storage Security in Cloud Computing", Proceedings of 2017 IEEE international conference on circuits and systems(ICCS2017).

[22] Nishit Mishra, Tarun Kumar Sharma, Varun Sharma and Vrince Vima, "Secure Framework for Data Security in Cloud Computing", ©Springer Nature Singapore Pvt. Ltd. 2018.

[23] Krishna Keerthi Ch,Lakshmi Muddan,Rajani Kanth A, "Performance Analysis of various Encryption Algorithms for usage in Multistage Encryption for Securing Data in Cloud", 2017 2nd IEEE International Conference On Recent Trends in Electronics Information &

Communication Technology (RTEICT), May 19-20, 2017.

[24] P.Y.A.Ryan,Preta, “Voter with Paillier encryption, Mathematical and computer modeling”, Elsevier pg.No 1646-16662,2008

[25] Mbarek Marwan, Ali Kartit and Hassan Ouahmane, “Applying Homomorphic Encryption For Securing Cloud Database” ,978-1-5090-0751-6/16/$31.00 ©2016 IEEE

[26] Radjab Harerimana,Syh-Yuan Tan and Wei-Chuen Yau, “A JAVA IMPLEMENTATION OF PAILLIER HOMOMORPHIC ENCRYPTION SCHEME”, 2017 Fifth International Conference on Information and Communication Technology (ICoICT). ISBN: 978-1-5090-4911-0 (c) 2017 IEEE

[27] Si Chen, Lin Li, Wenyu Zhang, Xiaolin Chang, Zhen Han,BOPE: Boundary order-preserving encryption scheme in relational database system, IEEE Open Access Journal 2017.

[28] Rivest R.L. (1993) Cryptography and machine learning. In: Imai H., Rivest R.L., Matsumoto T. (eds) “Advances in Cryptology “, ASIACRYPT '91. Lecture Notes in Computer Science, vol 739. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-57332-1_36

[29] Hacigümüs H., Mehrotra S. (2004) Performance-Conscious Key Management in Encrypted Databases. In: Farkas C., Samarati P. (eds) Research Directions in Data and Applications Security XVIII. IFIP International Federation for Information Processing, vol 144. Springer, Boston, MA. https://doi.org/10.1007/1-4020-8128-6_7

[30] H. Hacigumus, B. Iyer and S. Mehrotra, "Providing database as a service," Proceedings 18th International Conference on Data Engineering, pp. 29-38, 2002, https://doi.org/10.1109/ICDE.2002.994695.

[31] Arvind Arasu, Spyros Blanas, Ken Eguro, ManasJoglekar, Raghav Kaushik, Donald Kossmann, Ravi Ramamurthy, PrasangUpadhyaya, and RamarathnamVenkatesan. 2013. Secure database-as-a-service with Cipherbase. In Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. Association for Computing Machinery, New York, NY, USA, 1033–1036. DOI: https://doi.org/10.1145/2463676.2467797

[32] Fizza Shahid1, Humaira Ashraf1, Anwar Ghani 1,Shahbaz Ahmed Khan Ghayyur1,Shahaboddin Shamshirband 2,And Ely Salwana 3,” PSDS–Proficient Security Over Distributed Storage: A Method for Data Transmission in Cloud” , IEEE Access, VOLUME 8,Pg.No 118285-118295(2020)